

PAT-NO: JP409016521A
DOCUMENT-IDENTIFIER: JP 09016521 A
TITLE: PARALLEL BATCH PROCESSING SYSTEM
PUBN-DATE: January 17, 1997

INVENTOR-INFORMATION:
NAME
KOIZUMI, NORIYOSHI
UMEDA, HARUMASA

ASSIGNEE-INFORMATION:
NAME COUNTRY
N T T DATA TSUSHIN KK N/A
NIPPON TELEGR & TELEPH CORP <NTT> N/A

APPL-NO: JP07165900
APPL-DATE: June 30, 1995

INT-CL (IPC): G06F015/00, G06F001/00 , G06F013/00 ,
H04L012/40

ABSTRACT:

PURPOSE: To deliberately and efficiently execute plural batch jobs in a client- server type system.

CONSTITUTION: A schedule information generating part 15 which allocates the plural batch jobs to individual client terminal and also, generates schedule information including the running condition of each batch job and its execution schedule time, and an execution managing table 17 in which generated schedule information is stored at every client terminal are provided in a server

(19)日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-16521

(43)公開日 平成9年(1997)1月17日

(51)Int.Cl. ^a	識別記号	庁内整理番号	F I	技術表示箇所
G 0 6 F 15/00	3 1 0	9364-5L	G 0 6 F 15/00	3 1 0 M
1/00	3 7 0		1/00	3 7 0 C
13/00	3 5 5	9460-5E	13/00	3 5 5
H 0 4 L 12/40			H 0 4 L 11/00	3 2 1

審査請求 未請求 請求項の数3 O L (全 8 頁)

(21)出願番号 特願平7-165900

(22)出願日 平成7年(1995)6月30日

(71)出願人 000102728

エヌ・ティ・ティ・データ通信株式会社

東京都江東区豊洲三丁目3番3号

(71)出願人 000004226

日本電信電話株式会社

東京都新宿区西新宿三丁目19番2号

(72)発明者 小泉 徳善

東京都江東区豊洲三丁目3番3号 エヌ・

ティ・ティ・データ通信株式会社内

(72)発明者 梅田 晴正

東京都千代田区内幸町一丁目1番6号 日

本電信電話株式会社内

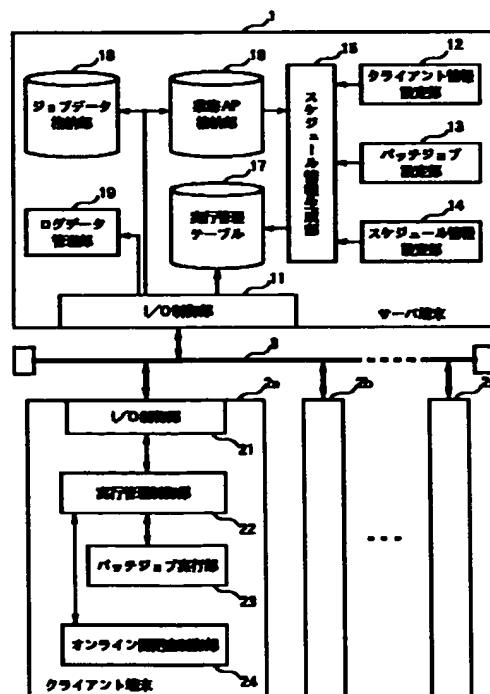
(74)代理人 弁理士 鈴木 正剛

(54)【発明の名称】 並列バッチ処理方式

(57)【要約】

【目的】 クライアント・サーバ型システムにおいて、複数のバッチジョブを計画的かつ効率的に実行する。

【構成】 複数のクライアント端末2a~2nが接続されたサーバ端末1に、複数のバッチジョブを個々のクライアント端末に割り当てるとともに、各バッチジョブの走行条件及びその実行予定時刻を含むスケジュール情報を生成するスケジュール情報生成部15と、生成したスケジュール情報をクライアント端末毎に格納した実行管理テーブル17とを備えておく。また、クライアント端末2a~2nに、各々実行管理制御部22とジョブ実行部23とを備え、実行管理テーブル17に格納された自端末宛のバッチジョブをその走行条件等にしたがって並列的に自動実行するようにした。



【特許請求の範囲】

【請求項1】 サーバ端末に複数のクライアント端末が接続されたクライアント・サーバ型システムにおけるバッチジョブの処理方式であって、

前記サーバ端末は、

複数のバッチジョブを個々のクライアント端末に割り当てるとともに、各バッチジョブの走行条件及びその実行予定時刻を含むスケジュール情報を生成するスケジュール情報生成部と、生成したスケジュール情報をクライアント端末毎に格納した実行管理テーブルと、を備え、

各クライアント端末は、

前記実行管理テーブルに格納されたスケジュール情報を取得する第1の手段、及び取得した実行スケジュール情報に含まれる自端末宛のバッチジョブの起動可否を判定する第2の手段を少なくとも備えた実行管理制御部と、前記バッチジョブの起動が可能で且つ該バッチジョブの実行予定時刻に達した時点で該バッチジョブをその走行条件に従って起動実行するバッチジョブ実行手段と、を有することを特徴とする並列バッチ処理方式。

【請求項2】 前記実行管理部は、前記実行管理テーブルに格納された自端末宛のスケジュール情報を更新する第3の手段を備えることを特徴とする請求項1記載の並列バッチ処理方式。

【請求項3】 前記サーバ端末は、ログデータ（バッチジョブにより加えられた変分情報を時系列に表したデータ）を各端末が随時参照可能に格納するログデータ管理部を有し、前記実行管理部は、バッチジョブの実行中に発生した前記ログデータを前記ログデータ管理部に時系列に格納する第4の手段を備えることを特徴とする請求項1又は2記載の並列バッチ処理方式。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、クライアント・サーバ型システムにおける運用形態を支援する技術に係り、特に、サーバ端末が割り当てた複数のクライアント端末により複数のバッチジョブを並列的に処理する方式に関する。

【0002】

【従来の技術】従来、ホストマシンを中心とした運用系処理は、全てミドルウェアパッケージのような、高機能マシンを前提としたパッケージにより支援されている。また、バッチジョブ走行に関しては、サーバ端末自体が一括処理するか、サーバ端末以外の複数の端末が処理したジョブ結果をサーバ端末が統括管理する方式を採用するのが一般的である。

【0003】

【発明が解決しようとする課題】近年、ダウンサイジング等により、既存のホスト系モデルの処理環境を維持・拡張しながら規模を縮小したクライアント・サーバ型システムの開発が進められている。このようなクライアン

ト・サーバ型システムでは、開発に際してホスト系モデルが備えている機能をいかに効率的に実現するかが重要な要素となる。しかし、現在のクライアント・サーバ型システムの開発支援環境において、ホスト系と同等の運用形態を支援する手段は提案されていないので、従来のミドルウェアパッケージのような機能を、相当の工数を費やしながら新規に開発する必要があった。中でも、複数のバッチジョブに関しては、従来より単なる縦並びのジョブ実行を支援しているパッケージは種々存在するものの、走行条件等を踏まえて効率的に運用系処理を支援することは行われていなかった。

【0004】本発明の課題は、かかる背景に鑑み、ダウンサイジング等における、クライアント・サーバ型システムのシステム開発支援環境において、従来のホスト系と同等の運用形態を支援する技術を提供することにある。

【0005】

【課題を解決するための手段及び作用】本発明は、ハードウェア技術の向上によりサーバ端末とクライアント端末の性能の差が縮小されており、また、サーバ端末に複数接続されるクライアント端末が常時稼働しているとは限らないという点に着目し、クライアント端末を有効に活用して複数のバッチジョブ処理の効率化を図ろうとしたものである。

【0006】具体的には、サーバ端末に、複数のバッチジョブを個々のクライアント端末に割り当てるとともに各バッチジョブの走行条件及びその実行予定時刻を含むスケジュール情報を生成するスケジュール情報生成部と、生成したスケジュール情報をクライアント端末毎に格納した実行管理テーブルとを備えておく。また、必要に応じて、ログデータを各端末が随時参照可能に格納するログデータ管理部をも備えておく。一方、各クライアント端末には、前記実行管理テーブルに格納されたスケジュール情報を取得する第1の手段、及び取得した実行スケジュール情報に含まれる自端末宛のバッチジョブの起動可否を判定する第2の手段を少なくとも備えた実行管理制御部と、前記バッチジョブの起動が可能で且つ該バッチジョブの実行予定時刻に達した時点で該バッチジョブをその走行条件に従って起動実行するバッチジョブ実行手段とを備えておく。この場合、前記実行管理部が前記実行管理テーブルに格納された自端末宛のスケジュール情報を更新する第3の手段をさらに備える構成であってもよく、また、前記サーバ端末に、ログデータ（バッチジョブにより加えられた変分情報を時系列に表したデータ）を各端末が随時参照可能に格納するログデータ管理部を設けるとともに、前記実行管理部に、バッチジョブの実行中に発生した前記ログデータを前記ログデータ管理部に時系列に格納する第4の手段を備える構成にしてもよい。

【0007】スケジュール情報には、上記バッチジョブ

の走行条件や実行予定時刻のほか、システム稼働日におけるオンライン開始/終了時刻等を含ませることができる。また、例えば日次、週次、月次、旬次、年次等の基準周期毎に生成しておくこともできる。

【0008】クライアント端末は、実行管理制御部の第1の手段で前記スケジュール情報を取得し、自端末宛のスケジュール情報が存在するか否かを確認する。そして、自端末宛のものが存在する場合は、第2の手段でバッチジョブの起動可否を判定する。この起動可否は、例えばオンライン終了時刻を経過しているか否かで判定することができる。すなわちデータの矛盾発生を防止するために、オンライン終了時刻を契機にオンラインを閉塞してバッチジョブの起動を可能とし、オンライン開始時刻を契機にオンライン閉塞を解除してバッチジョブの起動を不可能にする。バッチジョブの起動が可能となるときは、バッチジョブ実行手段が、スケジューリングされたバッチジョブをその走行条件に従って実行する。

【0009】なお、各クライアント端末に関わるスケジュール情報は、当該クライアント端末の実行制御部(第3の手段)から任意に修正(変更、追加等)することができ、前もって修正事項が判っている場合は、該修正後のスケジュール情報を予約することができる。また、バッチジョブの実行中にログデータをサーバ端末のログデータ管理部に格納し、故障発生時に随時これを参照するようにすることもできる。このログデータ管理部は、どの端末からも参照可能なので、どのクライアント端末で何のバッチジョブが何時何分に開始され、何時何分に終了したか、正常に終了しているかどうかの確認も可能になる。

【0010】

【実施例】次に、図面を参照して本発明の実施例を詳細に説明する。図1は、本発明の一実施例に係るオンラインシステムの構成図であり、一つのサーバ端末1と複数のクライアント端末2a~2nとをLAN等の通信路3を介して接続して成る。

【0011】サーバ端末1は、通信路3とのインタフェースとなるI/O制御部11のほか、クライアント情報設定部12、バッチジョブ設定部13、スケジュール情報設定部14、スケジュール情報生成部15、業務AP格納部16、実行管理テーブル17、ジョブデータ格納部18、及びログデータ管理部19を備えて構成される。一方、クライアント端末2a~2nは、それぞれ通信路3とのインタフェースとなるI/O制御部21のほか、実行管理制御部22、バッチジョブ実行部23、及びオンライン開閉塞制御部24を備えて構成される。なお、第1クライアント端末2aのみの構成を示しているが、他のクライアント端末2b~2nについても基本的には同様の構成となる。

【0012】まず、サーバ端末1の構成を具体的に説明する。クライアント情報設定部12は、システムで使用

する全クライアント情報の登録と、登録したクライアント端末毎のバッチジョブ開始時刻等を設定するものであり、バッチジョブ設定部13は、各クライアント端末に割り当てるバッチジョブを決定するとともに、システムの営業日情報、営業日におけるオンライン開始/終了時刻、日付変更時刻、休業日等を設定するものである。また、バッチジョブの起動周期、すなわち日次、週次、旬次、月次、季次、年次での詳細情報(実施日、曜日、月、間隔、周期が休日にあたった場合の処理等)をも必要に応じて設定することができる。スケジュール情報設定部14は、どの起動周期で何のバッチジョブ(起動業務AP)を起動するか、そのバッチジョブを走行させるクライアント端末はどこか、エラー時の後処理(リトライ、中断、次処理へスキップ等)をどうするか、オンライン中での実行可否、集計処理のクリア/繰り越しの有無、バッチジョブの自動/手動指定、バッチジョブの走行条件(何のジョブが終わっていないと実行できない等の指定)等を設定するものである。

【0013】スケジュール情報生成部15は、上記各種設定情報に基づいて実行管理テーブル17内に日次スケジュールテーブルのデフォルト値を自動生成するものである。実行管理テーブル17の詳細については後述する。業務AP格納部16は、各クライアント端末2a~2nが随時参照可能なバッチジョブ起動用の業務プログラムを格納しておくものであり、ジョブデータ格納部18は、各クライアント端末2a~2nが実行したジョブデータを格納するものである。なお、業務AP格納部16とジョブデータ格納部18は一体構成であってもよい。ログデータ管理部19は、各クライアント端末2a~2nから送られるログデータを、サーバ端末1を含め、全てのクライアント端末2a~2nからも参照可能に格納するものである。このログデータは、どのクライアント端末で何のバッチジョブが何時何分に開始され、何時何分に終了したかをも表すものである。

【0014】次に、実行管理テーブル17について説明する。この実行管理テーブル17はシステムに一つだけ存在し、ホスト端末1のほか、各クライアント端末2a~2nからも随時参照、更新可能なものである。その構造は、図2に示すように、バッチジョブの走行日及び登録クライアント毎に日次スケジュールテーブル17A~17Nが複数格納されている。このように、複数のものを図示しているのは、上述のようにサーバ端末1で生成するほか、各クライアントA~N側から予約等により生成できることを意味している。日次スケジュールテーブル17A~17Nのパートは、タイムテーブル部とクライアント情報部とバッチシーケンス部とに分かれ、各々情報のリンクがなされている。各部に格納される情報の一例、及び各部間のリンク状態を図3に示す。

【0015】図2及び図3に示すように、タイムテーブル部は、日次スケジュールテーブルに一つだけ存在し、

5

そのテーブル全体で利用する情報、例えば、バッチジョブの走行日や、この走行日とリンクして読み出したオンライン開始/終了時刻、日付切替時刻等がデフォルトで設定されている。

【0016】また、クライアント情報部は、個々の日次スケジュールテーブルに登録クライアント個数分存在して、クライアント固有の情報が随時修正可能に設定されている。そして、タイムテーブル部の「走行日」とクライアント情報部の「走行日」が一致するものを取得することで、設定されたクライアント情報を取得することが

できるようにになっている。
【0017】バッチシーケンス部は、クライアント単位にスケジュール項目数分存在し、バッチシーケンス順（周期順）にソートされている。クライアント情報部の「走行日」、「クライアント識別（図3の例ではクライアントA）」とバッチシーケンス部の「走行日」、「クライアント識別」の一致するものを取得することで、クライアント単位の日次スケジュール項目を取得できるようになっている。

【0018】図4は、このような設定内容の日付スケジュールテーブルの内容を、図示しないモニタ手段によってモニタする場合の表示例であり、クライアント端末Aについて索出した日次スケジュール項目の内容が示されている。これらの項目は、サーバ端末1がデフォルトで設定したものであるが、クライアントA側から、変更、追加、削除、移動を随時選択的に行うことができる。

【0019】次に、図1、図5及び図6を参照して、クライアント端末2aの構成について説明する。実行管理制御部22は、サーバ端末1との間でバッチジョブの起動実行に関するデータの授受を行うとともに、自端末内の統括制御を司るものであり、バッチジョブ実行部22は、自端末宛のバッチジョブを実行するものである。オンライン閉塞制御部24は、バッチジョブの実行に先だってオンラインを閉塞し、バッチジョブの実行終了後にオンライン閉塞を解除するものである。

【0020】実行管理制御部22の具体的なモジュール構成を図5に示す。この実行管理制御部22は、実行管理テーブル17、業務AP格納部16、ジョブデータ格納部18、及びログデータ管理部19にアクセス（情報参照、更新等）するサーバアクセス制御部221と、サーバ端末1より取得した情報に含まれるクライアント識別の内容を認識するクライアント識別部222と、オンライン開始/終了時刻や自端末宛のスケジュール情報がスケジューリングされている時刻に達しているか否かを監視する時間情報監視部223と、バッチジョブが現在どのくらい消化されているか（処理済み、未処理、実行中）を監視するジョブ走行監視部224と、バッチジョブの起動処理を行うバッチジョブ起動プロセス実行部225とをこの順に配するとともに、実行管理テーブル17に格納された自端末に関する日次スケジュールテ

6

ブル17A~17Nの内容の修正や予約等を、例えば図6に示す表示画面例に従って行うスケジュール設定部226を備えて構成される。

【0021】次に、上記構成のオンラインシステムにおいて、複数のクライアント端末2a~2nによるバッチジョブの実行手順を図7~図9に従って説明する。これらの図において、Sは処理ステップを表す。

【0022】図7は運用定義（情報設定）系の処理手順図であり、まず、サーバ端末1が、バッチジョブ名や実行開始時刻等の走行条件、オンライン開始/終了時刻、日付切替時刻等を取得し（S101）、各情報の編集後、各バッチジョブの走行日に関する日次スケジュールテーブル17Aを生成し（S102）、これを実行管理テーブル17に格納する（S103）。このときの日次スケジュールテーブル17の内容はサーバ端末1側が定めたデフォルト値であり、クライアント端末2a~2n側で修正する必要がある場合、あるいはクライアント端末2a~2nからスケジュール予約がない場合は定義系の処理を終える（S104）。なお、日付切替時刻に達したときは翌日分の日次スケジュールテーブルを生成する。

【0023】一方、クライアント端末2a~2nにおいて、修正を必要とする場合は該当項目を更新し（S105）、予約がある場合は予約テーブル（新たな日次スケジュールテーブル）を生成して実行管理テーブル17に格納する（S106）。上記更新や予約テーブルの生成はクライアント端末2a~2n側で行う。

【0024】次に実行系の処理手順を説明する。図8は、バッチジョブを実行するクライアント端末間における初期処理までの手順図である。該当クライアント端末の実行管理制御部22は、現在の自端末におけるジョブの実行状況と時間情報とを監視し（S201）、オンラインが閉塞時刻に達していることを確認した後（S202）、サーバ端末1にアクセスして実行管理テーブル17を参照する（S203）。なお、実行管理テーブル17の参照は、オンライン運用中でも可能であり、またオンラインの閉塞は手動によって任意の時刻に行うことができるが、本実施例では、便宜上それを考慮しないものとする。実行管理テーブル17を参照した結果、自端末宛にスケジュールされたものがあるか否かを判定し（S204）、自端末宛のものである場合は初期処理を行う（S205）。ここに、初期処理とは、例えばデータ保全のために行う、関連ファイルやテーブルのバックアップ処理や、その後のバッチジョブ起動プロセス実行のための前処理をいう。

【0025】初期処理実行後は、図9の処理手順に従ってバッチジョブを実行する。すなわち、バッチジョブ起動プロセスを実行し（S301）、日次スケジュール項目をゲットしたか否かを判定する（S302）。ゲットしたときは、自端末においてバッチジョブの起動実行が

可能か否かを判定し(S303)、実行不可の場合は初期処理実行後のプロセスに戻る(S303:実行不可)。実行可能の場合(S303:実行可)は、業務AP格納部16にアクセスして該当プログラムを読み出し、スケジューリングされた走行条件でバッチジョブを実行し、実行結果をジョブデータ格納部に順次格納するとともに(S304)、ログデータ管理部19へのログデータの書込を開始する(S305)。

【0026】ジョブが途中で中断、再開、強制終了のイベントが発生しないで終了したときは(S307)、ログデータの書込を終了させる(S308)。次のシーケンスのバッチジョブがあるときはS301の処理に戻り、ないときは終了時処理を行う(S315)。一方、S306において、中断イベントが発生したとき(S310)は、当該バッチジョブの実行を一時停止し(S313)、再開イベントが発生した時点で再開し(S314)、再度S306に戻る。S302で日次スケジュール項目をゲットできなかった場合(S302:No)、あるいは強制終了イベントが発生した場合(S312)は、直ちに終了時処理を行う。

【0027】このようにして、各クライアント端末2a~2bは、実行管理テーブル17に格納された日次スケジュールテーブル17を参照して各々自端末宛にスケジューリングされたバッチジョブを並列的に自動実行する。これにより、システム全体における複数のバッチジョブの処理時間が、サーバ端末1のみ、あるいはサーバ端末1を中心に行う場合に比べて格段に短縮され、ホスト系システムとほぼ同様の運用処理能力、運用形態を確保することができる。

【0028】また、実行管理テーブル17の格納情報をクライアント端末側から修正、予約することができ、さらにログデータ管理部19をどの端末からも参照できるようにしているので、個々のクライアント端末に何らかのトラブルが発生しても、即座に対応をとることができる。

【0029】

【発明の効果】以上の説明から明らかなように、本発明によれば、サーバ端末に設けた実行管理テーブルに複数のクライアント端末が随時アクセスし、該実行管理テーブルに格納されたバッチジョブのスケジュール情報にしたがって各々自端末宛のバッチジョブを並列的に自動実行するので、バッチジョブの処理効率が格段に高まる効果がある。また、従来のホスト系システムとほぼ同等の運用形態をサーバ端末側あるいはクライアント端末から

の修正、予約等による設定で実現できるため、運用形態プログラムの開発ないしその支援プログラムの開発が容易になる。

【図面の簡単な説明】

【図1】本発明の一実施例に係るオンラインシステムの構成図。

【図2】本実施例による実行管理テーブルの構造説明図。

【図3】本実施例による日付スケジュールテーブルの内容説明図。

【図4】日付スケジュールテーブルの設定内容を確認するときの表示例を示す図。

【図5】本実施例の各クライアント端末が備える実行管理制御部のモジュール構成図。

【図6】上記実行管理制御部が備えるスケジュール設定部の機能表示説明図。

【図7】運用定義系の処理手順図。

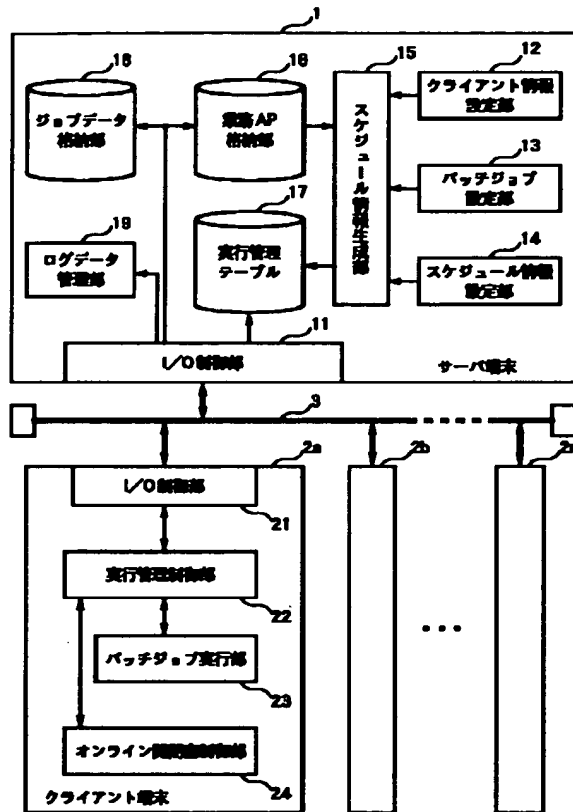
【図8】バッチジョブの実行時のクライアント端末の初期処理手順図。

【図9】初期処理後のクライアント端末におけるバッチジョブ実行手順図。

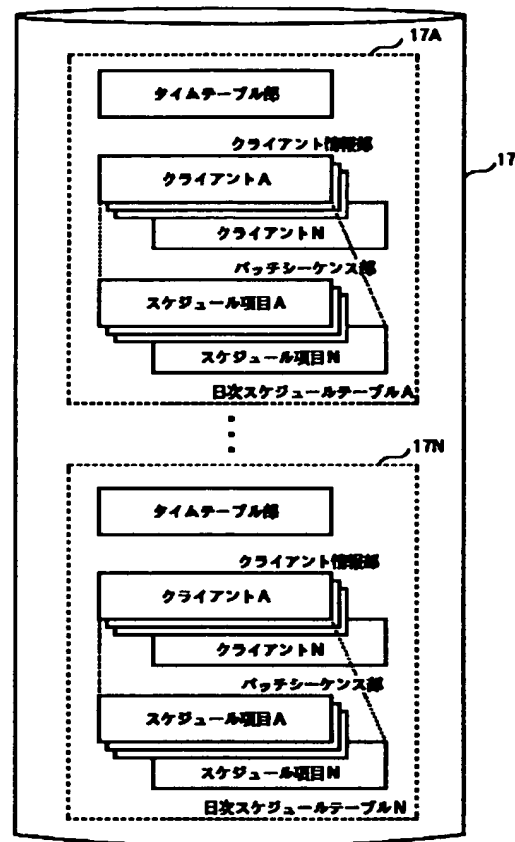
【符号の説明】

1	サーバ端末
11	I/O制御部
12	クライアント情報設定部
13	バッチジョブ設定部
14	スケジュール情報設定部
15	スケジュール情報生成部
16	業務AP格納部
17	実行管理テーブル
18	ジョブデータ格納部
19	ログデータ管理部
2a~2n	クライアント端末
21	I/O制御部21
22	実行管理制御部
221	サーバアクセス制御部
222	クライアント識別部
223	時間情報監視部
224	ジョブ走行監視部
225	バッチジョブ起動プロセス実行部
226	スケジュール設定部
24	オンライン開閉制御部
3	LAN等の通信路

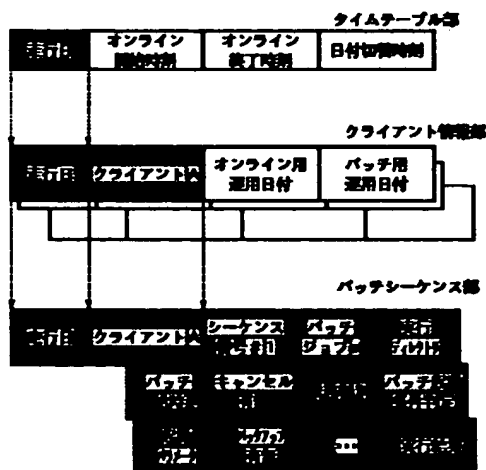
【図1】



【図2】



【図3】



【図4】

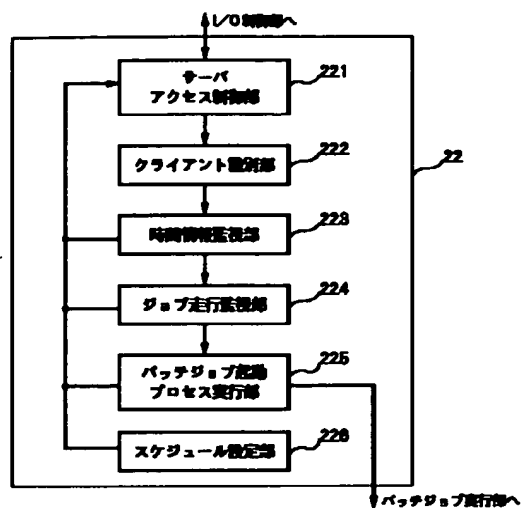
クライアントA 運用日付 Y1:M1:D1
バッチ開始時間 18:00

運用日付変更
バッチ開始時間変更

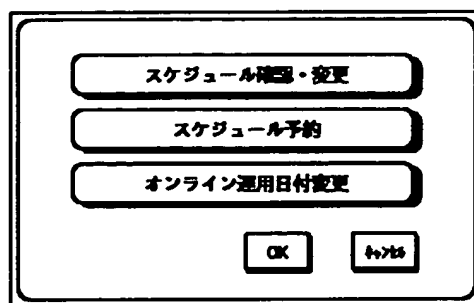
シーケンス番号	名称	周期	曜日	実行条件	エラー処理	実行中/実行済み
1	Y1A	日次	月曜		実行	しない
2	Y1B	日次	月曜		中断	しない
3	Y1C	週次	金曜	18:01 以降	中断	しない
...

変更 追加 削除 印刷 OK CAN

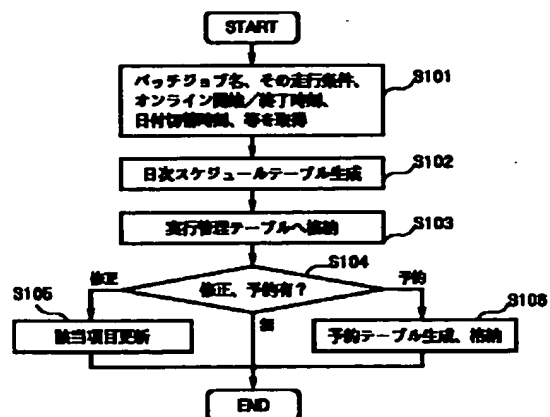
【図5】



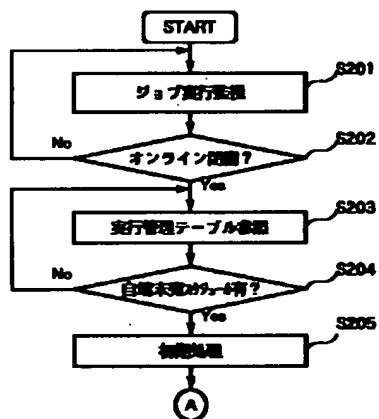
【図6】



【図7】



【図8】



【図9】

